PR4

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/477,688	01/06/2000	STEPHEN ANTHONY EDWARDS	1000/5	9589

35795 7590 09/10/2003

JONATHAN T. KAPLAN
ATTORNEY AT LAW
140 NASSAU STREET
NEW YORK, NY 10038-1501

EXAMINER	
ALI, SYED J	
ART UNIT	**PAPER NUMBER**
2127	15

DATE MAILED: 09/10/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

PRG

Office Action Summary	Application No.	Applicant(s)	
	09/477,688	EDWARDS, STEPHEN ANTHONY	
	Examiner	Art Unit	
	Syed J Ali	2127	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *15 July 2003* .

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-12* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-12* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *15 July 2003* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)☐ The proposed drawing correction filed on _____ is: a)☐ approved b)☐ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____ .

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) ☐ The translation of the foreign language provisional application has been received.

15)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449) Paper No(s) *13* .
4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: .

DETAILED ACTION

1. This office action is in response to Amendment A, paper number 12, which was received July 15, 2003. Applicant's arguments have been fully considered but they are not persuasive. Claims 1-12 are presented for examination.

2. The text of those sections of Title 35, U.S. code not included in this office action can be found in a prior office action.

Drawings

3. The drawings were received on July 15, 2003. These drawings are not acceptable. As noted on the initial PTO-948, drawings must be numbered consecutively. While it is acknowledged that a view labeled "8O" may be confused with the number "eighty", the MPEP is clear on the point that drawings must be numbered consecutively. Specifically, 37 CFR 1.84(u)(1) states:

> (u) Numbering of views.
> (1) The different views must be numbered in consecutive Arabic numerals, starting with 1, independent of the numbering of the sheets and, if possible, in the order in which they appear on the drawing sheet(s). Partial views intended to form one complete view, on one or several sheets, must be identified by the same number followed by a capital letter. View numbers must be preceded by the abbreviation "FIG. " Where only a single view is used in an application to illustrate the claimed invention, it must not be numbered and the abbreviation "FIG. " must not appear.

Claim Rejections - 35 USC § 102

4. Claims 1, 3, and 5-12 are rejected under 35 U.S.C. 102(b) as being clearly anticipated by Lin (see PTO-892 of previous Office action for citation).

As per claim 1, Lin discloses a method performed by a data processing system having a memory, comprising the steps of:

inputting a CCFG (pg. 213, fig. 2(a-b), 3(a-b), wherein the a CCFG is interpreted to be a collection of nodes and edges indicating program flow control distributed among a plurality of threads, and the figures in Lin depict two processes running in parallel, as claimed);

augmenting the CCFG with data edges to produce an augmented CCFG (pg. 212-213, fig. 2(a-b), 3(a-b), "Let G = <P, T, F, m_o> be a Petri net [14], where P is a set of places, T is a set of transitions", wherein it is well known in the art that when speaking of graph data structures, transitions between nodes [also known as places, states, etc.] are commonly referred to as edges);

scheduling the augmented CCFG to produce a scheduled augmented CCFG (pg. 215, "Our software synthesis method aims to produce, as intermediate output, plain C code that retains a high degree of parallelism so that the subsequent processor-specific code generation step can produce efficient executable machine code for the target processor", wherein the generation of compilable code is done specifically for the purpose of executing it on a processor, and thus inherently must be scheduled for execution);

selecting a first node of the scheduled augmented CCFG (pg. 213, fig 2(a-c), elements p1, p2, wherein copies of the nodes p1 and p2 are taken from the respective CCFGs in figs 2(a-b) and used as similar nodes in the SCFG of fig 2(c));

producing a first copy of the first node for an SCFG (see citation and remark above);

coupling, if a first thread of the first node is suspended, between a second node of the SCFG of a second previously-running thread and the first copy, a first context switch (fig. 2(c),

element c2, wherein the decision point c2 is coupled between the two concurrently executing threads, and is a point at which the flow of execution may switch contexts).

As per claim 3, Lin teaches the method of claim 1, wherein the translation of the CCFG into the SCFG produces, for each node of the CCFG, at most one corresponding node in the SCFG (pg. 213, "concurrent processes ca be composed via parallel composition...[,which is] essentially a Cartesian product of the two Petri net processes along common labeled actions", wherein the combination of more than one concurrent process is translated to a single sequential process, and the graphs are joined at places where data dependencies between the concurrent processes indicate that a context switch should take place. Therefore, each node in the original concurrent graph maps to at most one node in the sequential graph, the exception being where two states coincide and would thus collapse into one. The more common occurrence is that two transitions would coincide and that would indicate where a context switch should take place, in which case the nodes map on a one-to-one basis).

As per claim 5, Lin teaches the method of claim 1, wherein an execution of the SCFG comprises translation of the SCFG into a programming language (pg. 213-216, §4.2, "Our software synthesis method aims to produce, as intermediate output, plain C code that retains a high degree of parallelism so that the subsequent processor-specific code generation step can produce efficient executable machine code for the target processor", wherein it is clear that execution of the sequential data flow model results from the compilation of the expansion into C code and execution of that executable for a specific processor).

As per claim 6, Lin teaches the method of claim 5, wherein the programming language is C (pg. 213-216, §4.2, "Our software synthesis method aims to produce, as intermediate output, plain C code that retains a high degree of parallelism so that the subsequent processor-specific code generation step can produce efficient executable machine code for the target processor").

As per claim 7, Lin teaches the method of claim 1, further comprising a step of translation of the SCFG into a programming language (pg. 213-216, §4.2, "Our software synthesis method aims to produce, as intermediate output, plain C code that retains a high degree of parallelism so that the subsequent processor-specific code generation step can produce efficient executable machine code for the target processor").

As per claim 8, Lin teaches the method of claim 7, further comprising a step of executing the programming language translation of the SCFG (pg. 213-216, §4.2, "Our software synthesis method aims to produce, as intermediate output, plain C code that retains a high degree of parallelism so that the subsequent processor-specific code generation step can produce efficient executable machine code for the target processor").

As per claim 9, Lin teaches the method of claim 1, wherein an execution of the SCFG comprises interpretation of the SCFG (pg. 213-216, §4.2, "Our software synthesis method aims to produce, as intermediate output, plain C code that retains a high degree of parallelism so that the subsequent processor-specific code generation step can produce efficient executable machine

code for the target processor", wherein the translation into C code and generation of an executable therein amounts to interpretation of the SCFG resulting in an execution of said SCFG).

As per claims 10-12, essentially claim a data processing system having a memory, a computer program product comprising a computer readable medium having computer readable code embodied therein, and a computer data signal embodied in a carrier wave and representing sequences of instructions, respectively, each of which are adapted to perform the method of claim 1. To that end, the disclosure of Lin inherently includes each of these limitations. Specifically, Lin is disclosed as a way of scheduling tasks in an operating system, using a concurrent flow of execution to handle interprocess communication (Abstract). Furthermore, this technique is intended to be implemented in software, as Lin discloses pseudocode as well as sample C code that serves as an example of this. Therefore, Lin meets the limitations of claims 10-12, as discussed above, and the remainder of the limitations mirror those of claim 1, which is discussed above.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claim 2 is rejected under 35 U.S.C. 103(a) as being unpatentable over Lin in view of Nilsen et al. (USPN 6,081,665) (hereinafter Nilsen).

As per claim 2, Lin does not specifically disclose the method of claim 1, wherein the first context switch is achieved by adding code that saves a state of a thread being suspended in a state variable and resumes another thread by performing a multiway branch on a state variable for a thread being resumed.

However, the method of Lin does define data structures and a method that would allow a simple modification that could use state variables (pg. 213, fig. 2(c) and 3(c), wherein referring to figure 2(c), the transition point c2 is an exemplary instance of where a context switch may take place). For example, assume execution began at point p1. After transition point c2, execution may continue along one of two pathways. This comprises a multiway branch. There are various ways of suspending and resuming threads, and the use of state variables to indicate the status of a suspended or preempted thread is a well-known method and does not constitute an improvement over the prior art of Lin. One example of an implementation of threads using state variables can be found in the enclosed reference, Nilsen (col. 37, lines 60-67, "When a task is

preempted, the dispatcher saves this information into the thread's state variables. Upon resumption, the thread restores these state variables from the saved thread information"). It would have been obvious to one of ordinary skill in the art to use state variables for suspending and resuming threads since when multiple threads are running concurrently and context switches may occur between threads, there exists a strong possibility that when a context switch occurs, the suspended thread has not completed its task. Therefore, saving the state of the variables allows the thread to resume its functions at a later point without any loss of data.

7. Claim 4 is rejected under 35 U.S.C. 103(a) as being unpatentable over Lin.

As per claim 4, Lin does not specifically disclose the method of claim 1, wherein the step of scheduling further comprises a topological sort for determining the scheduled augmented CCFG.

"Official Notice" is taken that the use of topological sorts is well known and expected in the art, and would have been an obvious modification to Lin. Specifically, Lin discloses an ordered graph, with nodes used to refer to specific variable states and edges to traverse between those states (pg. 212-213, fig. 2(a-c), 3(a-c), wherein based on the code on pg. 212, a graphical representation is created in figures 2(a-b)).

A topological sort is well known as a way of ordering nodes, such that if a transition occurs between two nodes, then based on how that transition is represented, it is known what order the nodes occur in the graph. A common way of expressing this is that if an edge exists such that edge(u, v) is in the graph and u and v are nodes in the graph, then u comes before v in

the ordering of the graph. This can be found in any number of programming guides, and an example is presented in the Boost Graph Library on the Boost C++ Library website (www.boost.org/libs/graph/doc/topological_sort.html). It would have been obvious to one of ordinary skill in the art to use a topological sort to determine the ordering of the scheduled augmented CCFG since the technique is well known, other programming methods that have been previously devised can be used in accordance with the sorting technique. Specifically, defining the graph with similar data structures would allow a programmer a multitude of predefined methods to operate on the data therein, depending on the specific needs of each individual system.

Response to Arguments

8. Applicant argues on page 6, "*No...scheduled augmented CCFG is produced or suggested by the approach of Lin*" and "*No...conditional basis for the production of the SCFG is taught or suggested by Lin, which relies instead upon the production of a Petri net representation.*"

In response to the former argument, this limitation is addressed above, in reference to claim 1. The discussion therein is considered sufficient for showing how Lin does indeed discloses a scheduled augmented CCFG. Regarding the latter argument, that no conditional basis for production of the SCFG is taught or suggested by Lin, Examiner respectfully disagrees. Although Lin discloses a Petri net representation, this is merely one way of expressing a sequential control flow graph. By referring to Figs. 2(a-c), there can be no question that Lin uses conditional statements to convert two concurrent control flow graphs into a single sequential control flow graph. That is, conditional statements that may have data dependencies between

two separate threads are combined into branch points, at which context switches may occur. Clearly, Lin is using conditional statements, i.e., (x<0) in Fig. 2(c), to produce a sequential flow of control.

The remainder of the arguments presented merely alleges that claims 2-9 are allowable based on their dependence upon claim 1, and that claims 10-12 are allowable for being similar to claim 1. However, as claim 1 is taught by Lin, as discussed above under 102 Rejections, as well as in view of the Response to Arguments above, these claims are not patentable for at least the reasons presented for claim 1. Furthermore, discussion is presented as to how Lin discloses the limitations of each claim.

Conclusion

9. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Syed J Ali whose telephone number is (703) 305-8106. The examiner can normally be reached on Mon-Fri 8-5:30, 2nd Friday off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, William A Grant can be reached on (703) 308-1108. The fax phone number for the organization where this application or proceeding is assigned is (703) 872-9306.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

Syed Ali
August 28, 2003

WILLIAM GRANT
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

9/8/03